



COMPUTER
CENTRE
BULLETIN

Volume 1, Number 1.
1st July, 1968.

Editor:
J. S. Williams.

1. INTRODUCING THE COMPUTER CENTRE BULLETIN

1.1 What?

The COMPUTER CENTRE BULLETIN will be a monthly bulletin containing information about the Computer Centre and information relating to the use of the PDP 10 and GE 225 Computer Systems. It is hoped that the COMPUTER CENTRE BULLETIN will occasionally contain an article of general computer interest.

1.2 Why?

With the computer usage, here at the Computer Centre, expanding so rapidly it is felt by the staff that there is a growing need for some means of keeping clients informed about developments at the Centre. No longer is word of mouth an efficient enough method of informing clients on common programming problems, staff changes, recent publications, and new subroutines.

It is not intended that it be a large magazine but that it be more concerned with Computer Centre information and problems. The success of this venture will largely depend on the degree of co-operation of clients. If you have any article, item or topic that you feel would be of interest to other users please send it to the Editor and we will endeavour to circulate it through the COMPUTER CENTRE BULLETIN.

1.3 When and Where?

The COMPUTER CENTRE BULLETIN will be available at the Computer Centre on the first Monday of every month, except January and February. If any Organizations of individuals would appreciate having a copy mailed to them please fill in the application form in Section 5. Mailed copies should arrive, within the Brisbane area, approximately the same day as copies are available from the Computer Centre.

2. LIBRARY PROGRAMS

2.1 New Additions

The following mathematical and statistical subroutines have been added to the program library recently. All are in FORTRAN IV.

FORIF - Fourier analysis of periodic functions	(D6.206)
FORIT - Fourier analysis of periodic functions	(D6.208)
(The only difference between the above two is the method of supplying data to the subroutine)	
EXSMO - Triple exponential smoothing	(D3.252)
AUTO - Autocovariance of time series	(D3.254)
CROSS - Crossvariance of time series	(D3.256)

2.2 Future Additions

The programming staff at the Computer Centre are at present converting CARD FORTRAN programs and programs from other sources into FORTRAN IV. If there are requests for particular library programs, then an endeavour will be made to convert them in preference to other programs. Unfortunately, there can be no guarantee that conversion will be completed by a certain date, but an attempt will be made if the need is sufficiently great.

3. PROGRAMMING PROBLEMS AND USEFUL TECHNIQUES

An endeavour will be made to have this section at least a semi-regular part of the COMPUTER CENTRE BULLETIN. If any client knows of some point relating to programming, here at the Computer Centre, that requires clarification please put it in writing and send it to the Editor.

3.1 System Errors

Rapid notification of any errors that do not appear to lie in the user's program is appreciated. Such errors rarely occur but when one does it must be corrected as soon as possible. A small test program demonstrating the error is welcomed.

3.2 Consultation

The Computer Centre provides a consultation service to clients, at no charge, between the hours of 9 a.m. and 11 a.m. and 2 p.m. and 3 p.m., Monday to Friday. One member of the programming staff is on duty at any given time during the above hours. The roster for the week is posted on the Notice Board and may change from week to week.

The member of staff on duty can be contacted by ringing extension 690, or by asking the Administrative Officer. If a client wishes to consult a specific member of staff who is not on duty, he may ring the Secretary (ph. ext. 688) and make an appointment.

3.3 FORTRAN IV Execution Error Messages

The following execution error messages have been *added* to FORTRAN IV.

DIV	Attempt to divide by zero. Go to next job.
DSS	Floating point input datum having decimal scale of absolute value > 127.
ID1	Double precision involution, base zero, negative exponent. Result largest number.
ID2	Double precision involution, base zero, zero exponent. Result zero.
ID3	Double precision involution, negative base. Result positive base.
LBF	Error in DMOD routine. ARG1/ARG2 > 10**16.
LER	Logical input data in error, Go to next job.
SSS	Scale factor having absolute value > 127 on F or G conversion. Result zero for negative scale; else, largest positive number.

3.4 Variable Dimensions - FORTRAN IV

e.g. SUBROUTINE SBR(A, N, M)

DIMENSION A(N, M)

....

....

The importance of variably dimensioned 2D and 3D arrays is the fact that subroutines do not need recompiling to change the dimension statement. (With

a one dimensional array it is satisfactory to use dimension 1 in a subroutine for an array coming through as an argument). It is important that clients understand how to use this feature of FORTRAN IV, as most library subroutines from now on will use it.

If the difference in storage methods for fixed vs variable dimensions is understood, then it should not be a problem for a client used to fixed dimensions to use variable dimensions. The reason why the difference in storage must be appreciated is the fact that *dimensions in a main program must be fixed. Variable dimensions must only be used in a subprogram.*

Since the dimension statement in a main program must be fixed, a fixed area is set aside for the array. In the case of a fixed dimension statement in a subprogram, storage is not set aside for the array, but a method of linkage with the area in the main program is set up and the values in the dimension statement are used in calculating subscripts.

The situation is similar to this with variable dimensions, except the size of the array is obtained indirectly and so can vary.

Arrays not in the calling sequence of a subprogram must not be variably dimensioned in that subprogram.

With the difference in storage methods any array manipulation in the main program must be carried out *with great care*, keeping in mind the storage method. It is much wiser to carry out manipulation (including I/O) in subprograms.

(a) Fixed Dimension Method of Storage

e.g. DIMENSION A(20, 20) in a subroutine, as well as in the main program.

Suppose the array stored in A for a particular run is 10 x 15, then each column is stored from the start of the respective columns set aside by the dimension statement (see Figure 1).

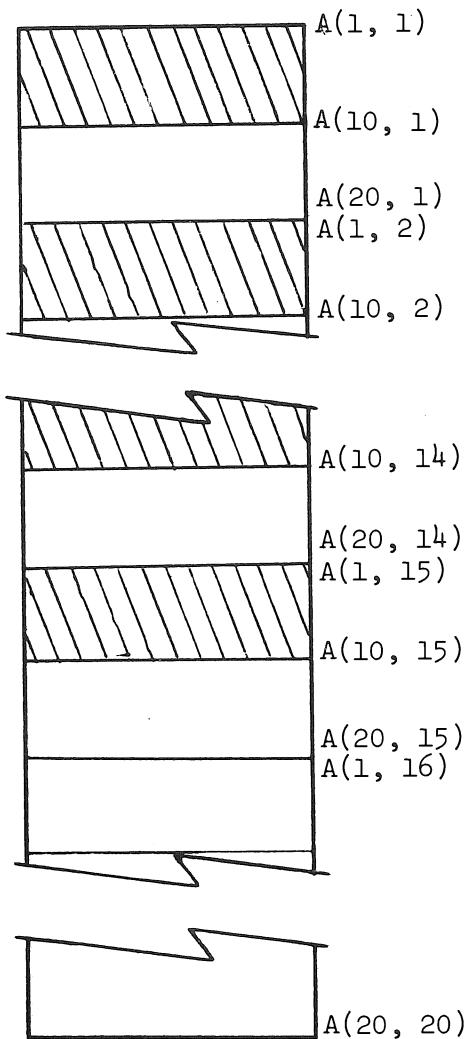


Figure 1. Fixed Dimension Method of Storage

The shaded area indicates the information stored.

(b) Variable Dimension Method of Storage

e.g. DIMENSION A(N, M) in a subroutine.

Suppose A is dimensioned 20×20 in the calling program, and for a particular run $N = 10, M = 15$. The 10×15 array is stored column by column with no consideration of the size stated in the main program (see Figure 2).

Of course, in the main program, information is expected to be stored as for fixed dimensions.

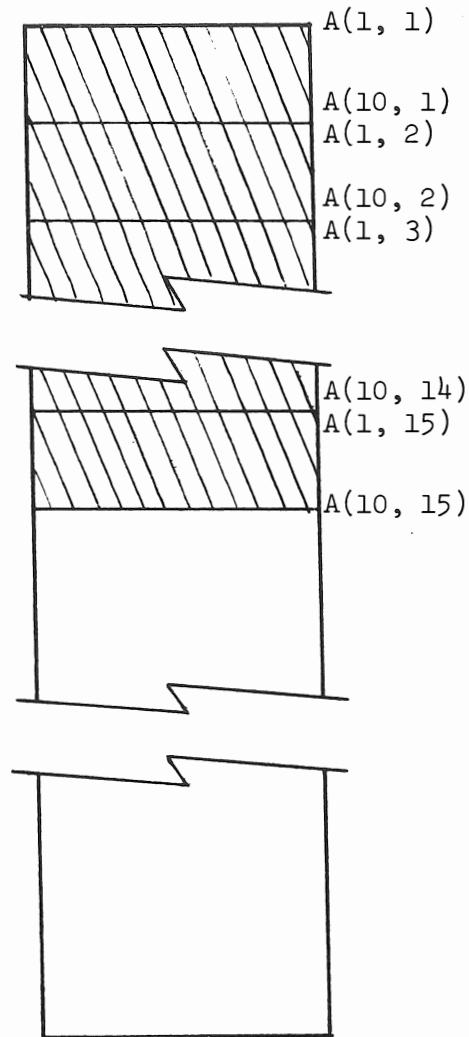


Figure 2. Variable Dimension Method of Storage

The shaded area indicates the information stored.

3.4 Submitting a Program or Data for Punching

It has been noted, recently, that a number of clients have not been adhering to the Computer Centre convention for the representation of certain characters, viz.

zero = 0	capital letter "o" = Θ
one = 1	capital letter "i" = I
two = 2	capital letter "z" = Z

Also care must be taken to distinguish between "5" and "S".

The Data Preparation Staff can not be expected to know which character is intended. Suppose a Programmer uses the variable name "FILE". He would not want it to be changed to "FILE". For this reason, and to avoid disappointment and confusion, the Data Preparation Staff have been given *strict instructions to punch "what is there"*.

Any programmer not adhering to the above character representation must suffer the consequences.

NOTE: All enquiries relating to Computer Services, including data preparation, are to be directed to the Administrative Officer. Ph. ext. 8471.

3.5 Carriage Control Characters in FORTRAN IV and CARD FORTRAN

The first character of each line printed by a PRINT or WRITE statement is suppressed if and only if it is one of the carriage control characters. (In FORTRAN IV these carriage control characters are 0, 1, and +; in CARD FORTRAN 0 and 1).

Contrary to popular belief, this is true of *all* printed lines and not just of lines beginning with a Hollerith string.

Example

```
IYEAR = 1968
PRINT 10, IYEAR
10   FORMAT (I4)
```

In the above example, a "1" appears in the first print position and hence is suppressed. The printer will slew to the top of the next page and print

1968

A safe policy is to begin the FORMAT description of each line with "1X".

Example

```
20 FORMAT (1X, I4, F12.5/1X, I6)
```

3.6 Miscellaneous

(a) It is not legal to have the combination ".-" in a logical IF, i.e. do not use a statement such as IF(A.GE.-5.0) GO TO 10.

This problem can be overcome by using brackets,

e.g. IF(A.GE.(-5.0)) GO TO 10

Similarly with the combination ".+".

(b) When using the EQUIVALENCE statement for arrays it is best to use constant subscripts (i.e. (A(1), B(1)) rather than (A, B)) otherwise there can be trouble. Remember that arrays are stored column by column and, with care, equivalencing the appropriate elements automatically equivalences the following elements of both arrays.

- (c) In a COMMON statement place all the integer variables before the real variables.
- (d) Take care not to inadvertently jump into a DO loop. When a program is compiled, coding for incrementing the index is placed *after* coding for the last statement in the DO loop. Therefore, even if the last statement is a CONTINUE statement it is still not legal to jump to it from outside the DO loop.
- (e) Variables to be placed in COMMON are assigned locations in COMMON in the order that they appear in the program and not always the order in the COMMON statement unless that is the first place they appear in the program.
- (f) The sequence - Logical IF - Computed GO TO is *not* compiled properly i.e. `IF(X.NE.0.0) GO TO (10, 20, 30), J`

4. RECENT PUBLICATIONS

The publications are available at the Computer Centre.

- (a) Technical Memorandum No. 1
University of Queensland Time-sharing System. E.J. Sokoll.
- (b) *Schedule of Charges for the PDP 10.*
- (c) Technical Manual No. 1
FORTRAN IV for the GE 225 and PDP 10. J.S. Williams.
- (d) Technical Memorandum No. 2.
GE 225 8-Bit N-Register. E.J. Sokoll.

5. APPLICATIONS

It is intended to keep two mailing lists, one for organizations and departments and the other for individuals. Since a larger proportion of individual clients are only closely associated with the Centre for a year or so, the individual mailing list will be kept on an annual basis.

The list for organizations and departments will be permanent. Prompt notification of any changes will be appreciated.

If you wish to receive copies please fill the appropriate section of the following form, sign and return it to:

The Editor,
Computer Centre Bulletin,
Computer Centre,
University of Queensland,
St. Lucia,
Queensland 4067

Application for Individual Circulation

Application for Organizational Circulation

NAME:

POSITION:

POSITION:

ORGANISATION:

ADDRESS:

ADDRESS:

.....

.....

.....

.....

NUMBER OF COPIES:

SIGNED: DATE:



